



# Цели и задачи проекта

- ▶ Разработать клиент-серверное приложение для обмена текстовыми сообщениями по сетям TCP/IP
- ▶ Реализовать его защиту
- ▶ Аудит безопасности приложения одногруппника

# Описание работы приложения

## Сервер

```
PORT: 5000
____Start Server____
secret key: MLNJAYTDC
```

При запуске сервера происходит проверка определённых записей в закомментированной области в оригинальном файле "hosts". В случае отсутствия этих записей сервер прекращает свою работу

При успешном прохождении проверки, сервер запрашивает порт, после этого, задается кодовое слово для авторизации клиента.



# Клиент

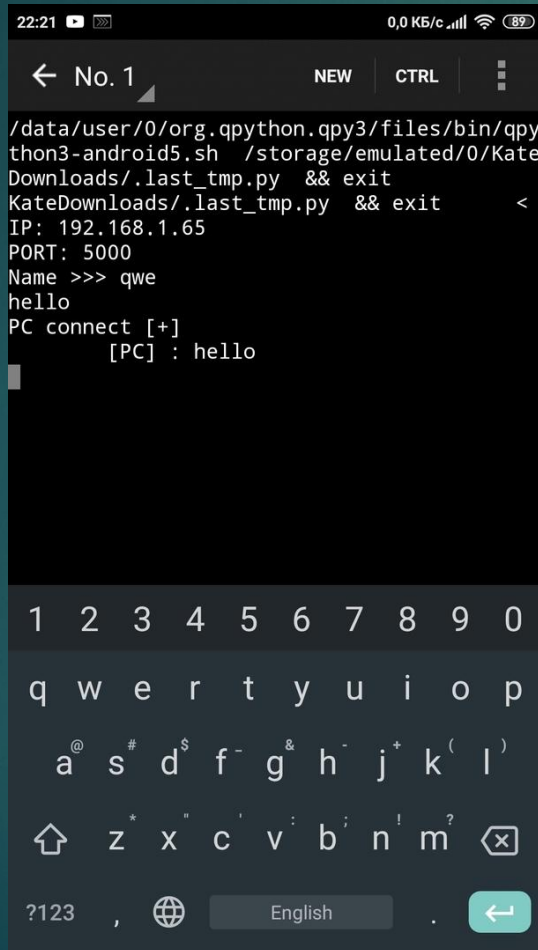
```
IP: 192.168.1.65
PORT: 5000
Name >>> qwe
      [caver] : hello
      [caver] : it`s client "one"
qwerty connect [+]
      [qwerty] : hello, it`s client "two"
```

При успешном прохождении проверки, пользователь должен ввести IP, порт, имя и ключ сервера, который генерируется при его запуске. При введении неправильного ключа, пользователь не попадает в чат.

При запуске клиента также проходит проверка наличия определённых записей в закомментированной области в оригинальном файле "hosts". В случае отсутствия этих записей клиент прекращает свою работу

Так же, пользователь сможет увидеть сообщения, которые были отправлены в чате до его появления.

# Запуск чата на нескольких устройствах



22:21 0,0 КБ/с 89

← No.1 NEW CTRL

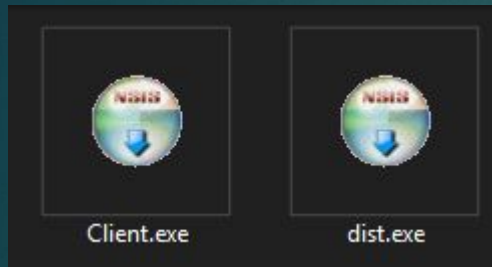
```
/data/user/0/org.qpython.qpy3/files/bin/qpython3-android5.sh /storage/emulated/0/KateDownloads/.last_tmp.py && exit
KateDownloads/.last_tmp.py && exit <
IP: 192.168.1.65
PORT: 5000
Name >>> qwe
hello
PC connect [+]
      [PC] : hello
```

1 2 3 4 5 6 7 8 9 0  
q w e r t y u i o p  
a @ s # d \$ f - g & h - j + k ( l )  
↑ z \* x " c ' v : b ; n ! m ? ⊞

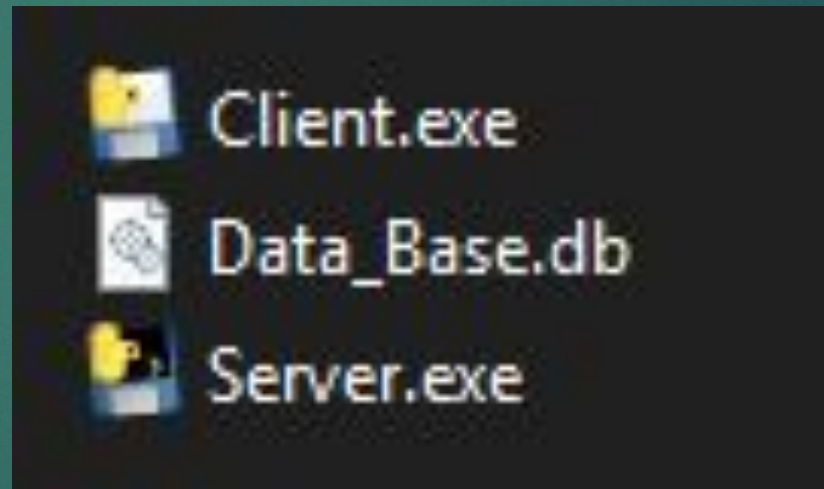
?123 , English ↵

```
IP: 192.168.1.65
PORT: 5000
Name >>> PC
      [qwe] : hello
hello
```

# Взлом оппонента



Изначально, я получил архив, с установочными файлами



После установки получаем 3 файла. Клиент, сервер и базу данных

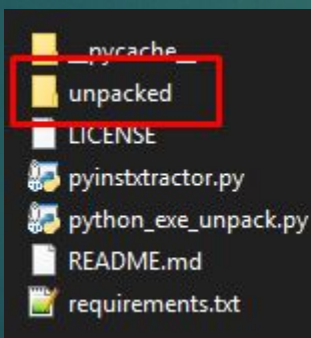


ВОСПОЛЬЗОВАВШИСЬ  
УТИЛИТОЙ  
python-exe-unpacker я  
распаковал .exe  
файлы

```
C:\Users\Caver\Desktop\dist\server\python-exe-unpacker-master>python_exe_unpack.py -i ..\Server.exe -o unpacked
[*] On Python 3.6
[*] Processing ..\Server.exe
[*] Pyinstaller version: 2.1+
[*] This exe is packed using pyinstaller
[*] Unpacking the binary now
[*] Python version: 37
[*] Length of package: 5337151 bytes
[*] Found 24 files in CArchive
[*] Beginning extraction...please standby
[!] Warning: The script is running in a different python version than the one used to build the executable
    Run this script in Python37 to prevent extraction errors(if any) during unmarshalling
[*] Found 136 files in PYZ archive
[*] Successfully extracted pyinstaller exe.
```

```
C:\Users\Caver\Desktop\dist\client\python-exe-unpacker-master>python_exe_unpack.py -i ..\Client.exe -o unpacked
[*] On Python 3.6
[*] Processing ..\Client.exe
[*] Pyinstaller version: 2.1+
[*] This exe is packed using pyinstaller
[*] Unpacking the binary now
[*] Python version: 37
[*] Length of package: 7195067 bytes
[*] Found 938 files in CArchive
[*] Beginning extraction...please standby
[!] Warning: The script is running in a different python version than the one used to build the executable
    Run this script in Python37 to prevent extraction errors(if any) during unmarshalling
[*] Found 135 files in PYZ archive
[*] Successfully extracted pyinstaller exe.
```

И ПОЛУЧИЛ  
ФАЙЛЫ



File Name	Date Modified	File Type	Size
PYZ-00.pyz_extracted	09.01.2020 2:05	Папка с файлами	
tcl	09.01.2020 2:48	Python Extension Module	72 КБ
tk	09.01.2020 2:48	Python Extension Module	32 КБ
_bz2.pyd	09.01.2020 2:48	Python Extension Module	181 КБ
_hashlib.pyd	09.01.2020 2:48	Python Extension Module	66 КБ
_lzma.pyd	09.01.2020 2:48	Python Extension Module	66 КБ
_socket.pyd	09.01.2020 2:48	Python Extension Module	103 КБ
_ssl.pyd	09.01.2020 2:48	Python Extension Module	769 КБ
_tkinter.pyd	09.01.2020 2:48	Python Extension Module	2 168 КБ
base_library.zip	09.01.2020 2:48	Сжатая ZIP-папка	525 КБ
Client	09.01.2020 2:48	Расширение при...	163 КБ
Client.bak	09.01.2020 2:48	Файл	5 КБ
Client.exe.manifest	09.01.2020 2:48	Файл	2 КБ
Client.pyc	09.01.2020 2:48	Файл	10 КБ
Client.pyc.bak	09.01.2020 2:48	Файл	19 КБ
Clientthack.pyc	09.01.2020 2:48	Файл "MANIFEST"	0 КБ
libcrypto-1_1.dll	09.01.2020 2:48	Расширение при...	3 522 КБ
libssl-1_1.dll	09.01.2020 2:48	Расширение при...	1 128 КБ
pyexpat.pyd	09.01.2020 2:48	Python Extension Module	163 КБ
pyiboot01_bootstrap	09.01.2020 2:48	Файл	5 КБ
pyimod01_os_path	09.01.2020 2:48	Файл	2 КБ
pyimod02_archive	09.01.2020 2:48	Файл	10 КБ
pyimod03_importers	09.01.2020 2:48	Файл	19 КБ
pyi-windows-manifest-filename Server.e...	09.01.2020 2:48	Файл "MANIFEST"	0 КБ
python37.dll	09.01.2020 2:48	Расширение при...	3 522 КБ




```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Текст декодирован
00000000 42 0D 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 В.....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 .....@.
00000020 00 73 5E 03 00 00 64 00 64 01 6C 00 5A 00 64 00 .s^...d.d.l.Z.d
00000030 64 01 6C 01 5A 01 64 00 64 02 6C 02 6D 03 5A 03 d.l.Z.d.d.l.m.Z
00000040 6D 02 5A 02 6D 04 5A 04 01 00 64 00 64 03 6C 05 m.Z.m.Z...d.d.l
00000050 6D 06 5A 06 01 00 64 00 64 01 6C 07 5A 07 64 04 m.Z...d.d.l.Z.d
```

Воспользовавшись Нех-редактором, я получил файл .рус и добавил заголовки, которые использовались другими .рус файлами

```
C:\Users\Caver\Desktop\dist\client\python-exe-unpacker-master\unpacked\Client.exe>uncompyl6 Client.py
# uncompyl6 version 3.6.2
# Python bytecode 3.7 (3392)
# Decompiled from: Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)]
# Embedded file name: Client.py
import hashlib, sys
```

После этого, я декомпилировал файлы .рус с помощью утилиты uncompyl6

И получил исходные файлы

 clienthack.py	09.01.2020 2:27	Файл "PY"	4 КБ
 Data_Base.db	09.01.2020 2:21	Data Base File	0 КБ
 serverhacked.py	09.01.2020 2:19	Файл "PY"	3 КБ



```
C:\Users\Caver\Desktop\dist\done\clienthack.py (done) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

serverhacked.py x clienthack.py x
1 import hashlib, sys
2 from socket import AF_INET, socket, SOCK_STREAM
3 from threading import Thread
4 import tkinter
5
6 def quit():
7     socket.send(bytes('/exit', 'utf8'))
8     sys.exit()
9
10
11 def log(event=None):
12     nickname = nic.get()
13     password = pas.get()
14     h = hashlib.md5(password.encode())
15     ht = h.hexdigest()
16     socket.send(bytes(nickname, 'utf8'))
17     socket.send(bytes(ht, 'utf8'))
18     msg = socket.recv(buff).decode('utf8')
19     if msg == '1':
20         login.destroy()
21     else:
22         label = tkinter.Label(login, text="Помока вхома.", fg="red")
23         label.place(x=28, y=98)
24
25
26 def conn(event=None):
27     global HOST
28     global PORT
29     HOST = host.get()
30     PORT = port.get()
31     connect.destroy()
32
33
34 def send(event=None):
35     msg = my_msg.get()
36     my_msg.set('')
37     socket.send(bytes(msg, 'utf8'))
38
39
40 def read_sock(event=None):
41     while True:
42         msg = socket.recv(buff).decode('utf8')
43         msg_list.insert(tkinter.END, msg)
44
45
46 connect = tkinter.Tk()
47 connect.title("Помока вхома")
48 connect.geometry("300x300")
49 connect.resizable(width=False, height=False)
50 host, port = tkinter.StringVar(), tkinter.StringVar()
51 label1 = tkinter.Label(connect, text="Хост:")
52 label2 = tkinter.Label(connect, text="Порт:")
53 entry1 = tkinter.Entry(connect, textvariable=host)
54 entry2 = tkinter.Entry(connect, textvariable=port)
55 button = tkinter.Button(connect, text="Отправить")
56 button.bind("<Button-1>", conn)
57 label1.place(x=10, y=20)
58 label2.place(x=22, y=58)
```

```
C:\Users\Caver\Desktop\dist\done\serverhacked.py (done) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

serverhacked.py x clienthack.py x
22
23 def process_conn():
24     while True:
25         client, client_address = SERVER.accept()
26         name = login(client)
27         print("%s: %s подключился" % client_address)
28         addresses[client] = client_address
29         Thread(target=processing_client, args=(client, name)).start()
30
31
32 def processing_client(client, name):
33     clients[client] = name
34     send(bytes("%s: подключение" % name, 'utf8'))
35     while True:
36         msg = client.recv(buff)
37         if msg == bytes('/exit', 'utf8'):
38             print("%s: отключился" % addresses[client])
39             client.close()
40             del clients[client]
41             break
42         elif msg == bytes('/register', 'utf8'):
43             if name != 'Admin':
44                 client.send(bytes("Нет доступа", 'utf8'))
45             else:
46                 connection = sqlite3.connect("data_base.db")
47                 cursor = connection.cursor()
48                 client.send(bytes("Введите никнейм", 'utf8'))
49                 nickname = client.recv(buff).decode('utf8')
50                 client.send(bytes("Введите пароль", 'utf8'))
51                 password = client.recv(buff).decode('utf8')
52                 h = hashlib.md5(password.encode())
53                 ht = h.hexdigest()
54                 zapros = "INSERT INTO login(Nickname, Password) VALUES('{}', '{}')".format(nickname, ht)
55                 cursor.execute(zapros)
56                 connection.commit()
57                 connection.close()
58                 client.send(bytes("Пользователь зарегистрирован", 'utf8'))
59             else:
60                 send(msg, name + ': ')
61
62
63 def send(msg, prefix=''):
64     for sock in clients:
65         sock.send(bytes(prefix, 'utf8') + msg)
66
67
68 clients = {}
69 addresses = {}
70 HOST = socket.gethostname(socket.gethostname())
71 PORT = 8080
72 print("Хост: " + HOST)
73 buff = 1024
74 print("Порт: " + str(PORT))
75 ADDR = (HOST, PORT)
76 SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
77 SERVER.bind(ADDR)
78 SERVER.listen(10)
79 print("Сервер запущен.")
```

Конец.

